

Modèle formel général pour le traitement d'interactions multimodales

Nadjet KAMEL⁺*

⁺ LRIA/USTHB
BP 32,

El Alia, Bab Ezzouar, 16111, Alger, Algerie
nkamel@wissal.dz , kamel@ensma.fr

Yamine AIT AMEUR^{*}

^{*} LISI/ENSMA
BP 40109

86961, Futuroscope Cedex, France
Yamine@ensma.fr

RESUME

Dans cet article nous nous intéressons à la modélisation de l'interaction multimodale en entrée. Un modèle formel d'interaction a été défini pour permettre la conception de tout type d'interface multi-modale. La sémantique du modèle est donnée à l'aide de systèmes de transitions étiquetées. Les propriétés de l'interface sont modélisées en logique temporelle CTL et vérifiées à l'aide du contrôleur sur modèle SMV.

MOTS CLES : Interactions Multimodales, Méthodes Formelles

ABSTRACT

This paper addresses the modeling of the multimodal interaction mainly input multimodal interactions. A formal model for the description of the user interaction is defined. The semantics of the model is given in terms of transition systems. The properties of the multimodal interface are specified in the CTL temporal logic and checked by the SMV model checker.

KEYWORDS : Multimodal Interaction, Formal Methods

INTRODUCTION

Avec la puissance des machines et le progrès des systèmes de reconnaissance de la parole et de synthèse vocale qui ont permis d'augmenter les capacités sensori-motrices et représentationnelles des systèmes, nous assistons au développement d'applications où les moyens de communication et d'interaction sont de plus en plus variés. Contrairement aux interfaces classiques où un seul mode d'interaction était offert, dans les interfaces multimodales l'utilisateur peut interagir en utilisant un ou plusieurs moyens de communication, avec l'application, tels que la parole, le geste et la manipulation directe etc. Dans ce type d'interfaces, les interactions peuvent être utilisées de manière séquentielle, parallèle, indépendante ou bien combinées de manière synergique.

Dans ce domaine, les travaux de recherche ont identifié plusieurs concepts liés aux systèmes multimodaux (mode, modalité, multimédia, multimodal...) sans atteindre un consensus sur les termes qui les désignent.

Nous retrouvons dans [17] une synthèse et une mise au point sur la terminologie et sur les référentiels applicables à la conception d'interfaces multimédias et multimodales. Dans ce domaine, plusieurs travaux de recherche s'intéressent à différents aspects. Certains s'intéressent à l'implémentation d'applications multimodales [7][16][3], d'autres à la définition de concepts relatifs à la multimodalité [17] et d'autres aux propriétés d'interfaces multimodales [8].

Les méthodes formelles, par l'utilisation de modèles et de techniques reposant sur des définitions mathématiques rigoureuses, offrent de nouvelles possibilités pour la validation des logiciels. Leur mise en application permet de garantir des propriétés fondamentales du système, avec pour résultat concret d'éliminer plus d'erreurs dès la conception. L'utilisation des techniques formelles dans le domaine des systèmes interactifs, que représente les IHM, a intéressé plusieurs travaux de recherche. Les premiers modèles ont utilisé des automates et/ou des extensions d'automates tels que les statecharts ou les ATN[24]. D'autres modèles se sont fondés sur des réseaux de Pétri [18] comme moyen de représentation du système interactif. Plus récemment, des extensions des automates (BDD) ont été utilisées comme modèle pour des logiques temporelles telles que CTL, CTL* [5] ou XTL [4]. Le système SMV[14] a été mis en oeuvre par [11] pour vérifier, par vérification sur modèle, des propriétés des IHM exprimées dans la logique CTL. Le langage synchrone LUSTRE a permis de vérifier et de générer du code dans les IHM à partir des descriptions écrites en Uil/X [23]. La spécification et la vérification d'IHM ont vu aussi l'utilisation des techniques algébriques. [20] furent les premiers à introduire la notion d'interacteur en utilisant LOTOS. En parallèle, d'autres techniques orientées modèle, basées sur la vérification d'obligations de preuve et sur la description incrémentale, ont été expérimentées sur les IHM comme Z[9], VDM[25] et B[1].

L'utilisation des techniques formelles dans le domaine des IHM multimodales (IHM3) reste rare et c'est dans ce contexte que se situe notre travail. Nous nous intéressons à l'utilisation des techniques formelles pour la modélisa-

tion de l'interaction homme machine en entrée avec une interface multimodale. Nous définissons un modèle formel de l'interaction multimodale en entrée. Ce modèle est défini dans l'objectif de permettre la conception d'interfaces multimodales à partir de différents types de multimodalité. La sémantique formelle de ce modèle est donnée en termes de systèmes de transitions étiquetées. Pour l'expression des propriétés des IHM3 nous utilisons une logique, en particulier une logique temporelle CTL. Pour ce travail, la représentation de ces systèmes de transitions ainsi que la vérification des propriétés sont assurées par le contrôleur sur modèle SMV. Les propriétés que nous exprimons puis vérifions sont des cas de propriétés CARE [8] identifiées pour les IHM3.

Cet article est structuré comme suit : la prochaine section et la section 3 présentent respectivement les IHM3 et leurs propriétés. La section 4 présente les références des travaux utilisant des techniques formelles pour les IHM3 ainsi que notre approche. Notre modèle d'interaction générique sera présenté dans la section 6. Dans la section 7, nous présentons un modèle paramétré selon le type de multimodalité de l'interface. Nous présentons une mise en œuvre de ce modèle sur l'étude de cas *Matis* dans la section 8. La dernière section présente la conclusion et les perspectives ouvertes par cette étude.

IHM MULTIMODALES (IHM3)

L'interaction dans les IHM3 est complexe dans le sens où elle est définie par plusieurs événements provenant de plusieurs canaux d'entrée, ce qui engendre un fort degré de parallélisme. Plusieurs modalités peuvent participer à la réalisation d'une interaction élémentaire : un *énoncé*. Ces énoncés sont composés pour produire des *tâches* et *sous tâches* d'interactions utilisateurs plus complexes. Les IHM3 peuvent être classées selon plusieurs critères combinés : ou type de multimodalité. Une classification, donnée dans [2], repose sur les trois critères suivants :

- production des énoncés : séquentiel ou parallèle ;
- usage des médias : exclusif ou simultané ;
- nombre de médias par énoncé.

Lorsque plusieurs médias sont utilisés pour un même énoncé, une fusion est nécessaire.

La combinaison de ces trois dimensions a produit sept types de multimodalité.

1. *Exclusive* : une seule modalité est utilisée par énoncé et la production des énoncés est séquentielle.
2. *Alternée* : plusieurs modalités peuvent être utilisées alternativement pour produire un énoncé. Les énoncés sont produits de manière séquentielle.
3. *Synergique* : les énoncés sont produits de manière séquentielle mais plusieurs médias peuvent être utilisés dans un même énoncé et de manière parallèle.

4. *Parallèle exclusive* : plusieurs énoncés indépendants peuvent être produits en parallèle. Un seul média est utilisé pour chaque énoncé et à un instant donné un seul média est actif.
5. *Parallèle simultanée* : plusieurs énoncés indépendants peuvent être produits en parallèle. Un seul média est utilisé pour chaque énoncé mais à un instant donné plusieurs médias peuvent être actifs en même temps.
6. *Parallèle alternée* : plusieurs énoncés indépendants peuvent être produits en parallèle. Plusieurs médias peuvent être utilisés dans un même énoncé mais un seul media est actif.
7. *Parallèle synergique* : plusieurs énoncés indépendants peuvent être produits en même temps. Plusieurs modalités peuvent être utilisées pour un même énoncé et plusieurs modalités peuvent être actives en même temps.

PROPRIETES DES IHM3

Plusieurs techniques et méthodes ont été définies pour tester l'utilisabilité des systèmes interactifs. Elles se fondent essentiellement sur l'identification des propriétés que le système doit vérifier et sur la sélection des méthodes et des outils permettant la prise en charge de l'évaluation des propriétés désirées. Bien que les propriétés liées à la validité et à la robustesse soient utiles pour caractériser les systèmes interactifs, elles ne couvrent pas complètement les spécificités des nouvelles technologies d'interaction telles que les IHM3. Afin de palier cette insuffisance, de nouvelles catégories de propriétés ont été définies pour les IHM3. Ce sont les propriétés dites CARE (Complementarity, Assignment, Redundancy, Equivalence). Une définition formelle de ces propriétés a été donnée dans [8]. Elle se base sur les notions d'*états*, *but*, *modalité* et *relations temporelles*.

UTILISATION DES APPROCHES FORMELLES POUR LES IHM3

Travaux Antérieurs

Peu de travaux mettant en œuvre des techniques formelles se sont intéressés à la modélisation et à la vérification et validation d'IHM3. Trois approches ont retenu notre attention. En premier, les travaux [21] utilisent les interacteurs pour modéliser l'interface de *Matis* à partir d'un modèle de tâche de *Matis* défini avec UAN. Les interacteurs ont été implémentés en Lotos. Un ensemble de propriétés a été vérifié à l'aide de la logique temporelle ACTL avec le contrôleur sur modèle de l'outil Lite. Ensuite, [10] et [12] montrent comment des techniques telles que Z et CSP peuvent être utilisées pour modéliser une application multimodale. Dans ce travail, les auteurs font abstraction des modalités, et évitent la prise en charge de toutes les spécificités d'une interface multimodale. Les problèmes de fusion et de vérification des propriétés CARE ne sont pas traités. Enfin, [19] utilisent le modèle ICO pour représenter la fusion de deux moda-

lités : le geste et la parole. Ce travail ne traite pas de la vérification de propriétés spécifiques aux IHM3.

Notre Proposition

Par rapport aux trois approches précédentes, notre démarche est générale. Elle permet la représentation de toute interaction multimodale en entrée. Le concepteur décrit les différentes interactions multimodales en utilisant des systèmes de transitions associés à tout type de modalités. Ensuite ces systèmes sont composés pour réaliser l'opération de fusion. Nous donnons dans la section suivante la formalisation de cette proposition.

Mise en Oeuvre dans un Outil de Vérification

Pour illustrer cette approche, nous avons utilisé le contrôleur sur modèle SMV, mais toute autre technique formelle supportant la représentation de systèmes de transitions aurait pu être retenue.

La vérification sur modèle, connue par modèle-checking, est une technique basée sur la construction d'un modèle fini du système et la vérification des propriétés sur ce modèle. La vérification est effectuée par énumération exhaustive de l'espace d'états. L'approche que nous utilisons dans cet article a été développée par Clarke et Emerson [6] et Quille et Sifakis [22]. Elle consiste à exprimer les propriétés dans une logique temporelle [13] et modéliser le système par un système de transitions fini. Un algorithme est utilisé pour vérifier si le système de transitions est un modèle pour la spécification.

En résumé, notre travail propose la modélisation de l'interaction en entrée dans une IHM3. La sémantique du modèle proposé est définie à l'aide des systèmes de transitions étiquetés. Les propriétés de l'interface sont modélisées en logique temporelle CTL et la vérification est effectuée à l'aide de la technique de vérification sur modèle en utilisant le contrôleur sur modèle SMV.

MODELE D'INTERACTIONS MULTIMODALES

Nous proposons de définir un modèle formel pour la modélisation de l'interaction utilisateur en entrée avec un système interactif multimodal. La syntaxe du modèle est donnée par des règles de grammaire et la sémantique des mots de cette grammaire est donnée en termes de systèmes de transitions étiquetés.

Nous définissons la grammaire BNF permettant de générer le modèle d'interaction multimodal par un ensemble de règles. La règle S est utilisée pour générer le langage en fonction des tâches utilisateurs à un niveau élevé d'abstraction. Ensuite, chaque tâche T est définie en fonction des sous tâches qui la composent jusqu'à sa définition de base à l'aide des énoncés. Enfin, chaque énoncé E est défini par la composition des événements élémentaires e qui le composent. L'ensemble des événements produits par l'ensemble des modalités du système est noté A .

Syntaxe

$S ::= S[]S \mid S>>S \mid S \parallel S \mid S \parallel\parallel S \mid T$

$T ::= T[] \mid T \mid T >> T \mid T \parallel T \mid T \parallel\parallel T \mid E$

$E ::= e;E \mid e \parallel E \mid e \parallel\parallel E \mid \delta$ avec $e \in A$

Soient P et Q deux termes de la grammaire et e un élément de l'ensemble A alors :

δ est un terme qui ne fait rien (mot vide) ;

$[]$ est le choix non déterministe entre deux tâches.

L'exécution de $P[]Q$ est soit celle de P, soit celle de Q ;

$>>$ est l'opérateur de séquence entre deux tâches.

L'exécution de $P>>Q$ est celle de P suivie de celle de Q ;

$;$ est le préfixage. L'exécution de $e;P$ est l'exécution de l'événement e suivie l'exécution de P ;

\parallel désigne l'opérateur du parallèle entrelacé. $P \parallel Q$ est la composition parallèle entrelacée des deux exécutions de P et Q (asynchrone);

$\parallel\parallel$ est l'opérateur de vrai parallélisme. $P \parallel\parallel Q$ est la composition parallèle entre les exécutions de P et Q (synchrone).

Sémantique des opérateurs

La sémantique formelle des différents opérateurs est donnée par des règles de transition définissant le système de transitions sous-jacent. Soient P, Q des termes de la grammaire, e, e_1 et e_2 des événements de A .

$P \xrightarrow{e} Q$ décrit le passage d'un état caractérisé par P à un état caractérisé par Q, lorsque l'événement e est déclenché. Les relations de transitions sont décrites selon le style de G.Plotkin [26] au moyen de règles de la forme

$\frac{\text{prémises}}{\text{conclusion}}$. Elles sont définies par deux axiomes et neuf règles.

$\delta : \delta \not\rightarrow$

$;; e;P \xrightarrow{e} P$

$[] : \frac{P \xrightarrow{e} P'}{P[]Q \xrightarrow{e} P'} \quad \frac{Q \xrightarrow{e} Q'}{P[]Q \xrightarrow{e} Q'}$

$>> : \frac{P \xrightarrow{e} P' \text{ et } P' \neq \delta}{P >> Q \xrightarrow{e} P' >> Q}$

$\frac{P \xrightarrow{e} P' \text{ et } P' = \delta}{P >> Q \xrightarrow{e} Q}$

$\parallel : \frac{P \xrightarrow{e} P'}{P \parallel Q \xrightarrow{e} P' \parallel Q} \quad \frac{Q \xrightarrow{e} Q'}{P \parallel Q \xrightarrow{e} P \parallel Q'}$

$\parallel\parallel : \frac{P \xrightarrow{e} P'}{P \parallel\parallel Q \xrightarrow{e} P' \parallel\parallel Q} \quad \frac{Q \xrightarrow{e} Q'}{P \parallel\parallel Q \xrightarrow{e} P \parallel\parallel Q'}$

$$\frac{P \xrightarrow{e_1} P' \text{ et } Q \xrightarrow{e_2} Q' \text{ avec } e_1 \in A_{mi}, e_2 \in A_{mj} \text{ et } A_{mi} \neq A_{mj}}{P \parallel Q \xrightarrow{(e_1, e_2)} P' \parallel Q'}$$

Le système de transitions d'un terme (mot) du langage généré par la grammaire est obtenu par induction structurée à partir des règles précédentes.

PARAMETRISATION DU MODELE D'INTERACTIONS MULTIMODALES :

Le modèle présenté dans la section précédente définit l'interaction multimodale de façon générique sans prendre en considération le type de multimodalité de l'interface. En effet, l'ensemble A contient des évènements que nous n'avons pas précisés. Il s'agit d'un paramètre assurant la généralité de notre proposition. Dans cette section nous proposons d'instancier le modèle afin de permettre la conception du modèle selon le type de multimodalité choisie.

Nous conservons les mêmes opérateurs ainsi que leur sémantique définis dans la section précédente. Les tâches et les énoncés sont paramétrés par les modalités utilisées dans leur réalisation. Notons A_{mi} , l'ensemble des événements ei produits par la modalité mi . Ainsi, si

on dispose de n modalités, on pose $A = \bigcup_{i=1}^n A_{mi}$ et on

note e les éléments de A . Enfin, AM désigne l'ensemble des évènements générés par un sous-ensemble de modalités possibles. Pour n modalités alors $AM = \bigcup_{k=1..n} A_{mk}$

avec $A_{mk} \subseteq A$.

A partir des définitions précédentes et du modèle sémantique précédent, nous pouvons décliner différents types d'interactions multimodales. Chaque type correspond à un sous-ensemble du modèle général de la section précédente. La sémantique reste toujours exprimée par le même type de système de transitions. Nous donnons ci-dessous différents types d'interactions multimodales bâties sur un ensemble d'évènements AM quelconque.

Type Exclusif

Les énoncés sont composés de manière séquentielle. Un énoncé est composé des évènements issus d'une seule modalité.

$$S ::= S [] S \mid S >> S \mid T_{AM}$$

$$T_{AM} ::= T_{AM} [] T_{AM} \mid T_{AM} >> T_{AM} \mid E_{Amk} \text{ avec } Amk \subseteq AM$$

$$E_{Ami} ::= ei ; E_{Ami} \mid \delta \text{ avec } ei \in A_{mi}$$

Type Alterné

Les énoncés composés séquentiellement et sont produits par des évènements issus de différentes modalités entrelacées.

$$S ::= S [] S \mid S >> S \mid T_{AM}$$

$$T_{AM} ::= T_{AM} [] T_{AM} \mid T_{AM} >> T_{AM} \mid E_{AM}$$

$$E_{AM} ::= e ; E_{AM} \mid e \parallel E_{AM} \mid \delta \text{ avec } e \in AM$$

Type Synergique

Les énoncés sont composés de manière séquentielle et peuvent être produits par plusieurs modalités en parallèle ou entrelacées.

$$S ::= S [] S \mid S >> S \mid T_{AM}$$

$$T_{AM} ::= T_{AM} [] T_{AM} \mid T_{AM} >> T_{AM} \mid E_{AM}$$

$$E_{AM} ::= e ; E_{AM} \mid e \parallel E_{AM} \mid e \parallel E_{AM} \mid \delta \text{ avec } e \in AM$$

Type Parallèle Exclusif

Chaque énoncé est composé d'évènements issus d'une seule modalité.

$$S ::= S [] S \mid S >> S \mid S \parallel S \mid T_{AM}$$

$$T_{AM} ::= T_{AM} [] T_{AM} \mid T_{AM} >> T_{AM} \mid T_{AM} \parallel T_{AM} \mid E_{Amk} \text{ avec } A_{mk} \subseteq AM$$

$$E_{Ami} ::= ei ; E_{Ami} \mid \delta \text{ avec } ei \in A_{mi}$$

Type Parallèle Simultané

Chaque énoncé est produit par les évènements issus d'une seule modalité. Plusieurs évènements issus de modalités différentes peuvent être déclenchés en même temps.

$$S ::= S [] S \mid S >> S \mid S \parallel S \mid T_{AM}$$

$$T_{AM} ::= T_{AM} [] T_{AM} \mid T_{AM} >> T_{AM} \mid T_{AM} \parallel T_{AM} \mid E_{Amk} \text{ avec } A_{mk} \subseteq AM$$

$$E_{Ami} ::= ei ; E_{Ami} \mid \delta \text{ avec } ei \in A_{mi}$$

Type Parallèle Alterné

Les énoncés peuvent être produits par plusieurs modalités en parallèle ou entrelacés.

$$S ::= S [] S \mid S >> S \mid S \parallel S \mid T_{AM}$$

$$T_{AM} ::= T_{AM} [] T_{AM} \mid T_{AM} >> T_{AM} \mid T_{AM} \parallel T_{AM} \mid E_{AM}$$

$$E_{AM} ::= e ; E_{AM} \mid e \parallel E_{AM} \mid \delta \text{ avec } e \in AM$$

Type Parallèle Synergique

Les énoncés sont composés de manière parallèle ou séquentielle et produits par plusieurs modalités. Plusieurs évènements peuvent être déclenchés en même temps.

$$S ::= S [] S \mid S >> S \mid S \parallel S \mid S \parallel S \mid T_{AM}$$

$$T_{AM} ::= T_{AM} [] T_{AM} \mid T_{AM} >> T_{AM} \mid T_{AM} \parallel T_{AM} \mid T_{AM} \parallel T_{AM} \mid E_{AM}$$

$$E_{AM} ::= e ; E_{AM} \mid e \parallel E_{AM} \mid e \parallel E_{AM} \mid \delta \text{ avec } e \in AM$$

Notre démarche est générique et a permis de représenter les différentes catégories de multimodalités identifiées. De plus, la sémantique pour chaque type est définie une fois pour toute. Un premier type de vérification de propriétés peut être réalisé par l'inclusion de langages.


```
Remplir1_To = « to Oslo » ;  $\delta$ 
Remplir2_To = Click_To ; Click_Oslo ;  $\delta$ 
```

2. La tâche *Formuler_requête* est exprimée en fonction des énoncés par l'expression suivante :

```
(Créer1_req [] Créer2_req )
>> (Remplir1_From [] Remplir2_From)
>> (Remplir1_To [] Remplir2_To)
```

3. La tâche *Formuler_requête* est exprimée en fonction des événements par l'expression suivante :

```
((« show me flights » ;  $\delta$ ) []
(Click_n_requête ;  $\delta$ ))
>> ((« From Boston » ;  $\delta$ ) []
(Click_From ; Click_Boston ;  $\delta$ ))
>> ((« To Oslo » ;  $\delta$ ) []
(Click_To ; Click_Oslo ;  $\delta$ ))
```

Le système de transitions de la tâche *Formuler_requête* est obtenu par la composition des systèmes de transitions des énoncés qui la composent. Le système résultant est décrit en figure 3.

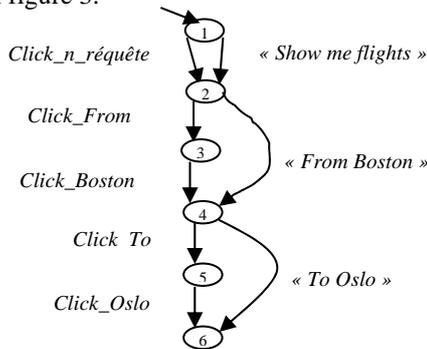


Figure 3 : Système de Transitions Etiquetées de la tâche Formuler_requête

Interface de type alterné. Plusieurs modalités participent dans la production d'un même énoncé mais une seule est active en même temps.

1. Les expressions de l'ensemble des énoncés sont :

```
Créer_requête1 = « Show me flight » ;  $\delta$ 
Créer_requête2 = Click_n_requête ;  $\delta$ 
Remplir_From = (« From this city » |||
Click_Boston) ;  $\delta$ 
Remplir_To = (« To this city » |||
Click_Oslo) ;  $\delta$ 
```

2. La tâche *Formuler_requête* est exprimée en fonction des énoncés par l'expression suivante :

```
(Créer_requête1 [] Créer_requête2)
>> ((Remplir_from >> Remplir_To)
[]
(Remplir_To >> Remplir_from))
```

3. La tâche *Formuler_requête* est exprimée en fonction des événements par l'expression suivante :

```
((« Show me flight » ;  $\delta$ )
[] (Click_n_requête ;  $\delta$ ))
>> (
((« From this city » ||| Click_Boston) ;  $\delta$ )
>>
((« To this city » ||| Click_Oslo) ;  $\delta$ )
[]
(((« To this city » ||| Click_Oslo) ;  $\delta$ )
>>
((« From this city » ||| Click_Boston) ;  $\delta$ ))
)
```

Le système de transitions correspondant contient 13 états numérotés de 1 à 13 et vu sa taille importante et l'espace limité dans cet article nous avons choisi de ne pas le présenter ici.

Vérification

Pour vérifier les propriétés CARE, il faut raisonner sur les modalités des actions effectuées. Pour cela, il est utile de remplacer les étiquettes des actions, dans les systèmes de transitions, par leurs modalités respectives. Cette opération est une interprétation abstraite. Elle est effectuée en abstrayant chaque étiquette d'une transition, qui correspond à l'action effectuée, par le type de sa modalité. Le système de transitions obtenu est une abstraction correcte du système d'origine (une sorte de typage). Ainsi le système de transition de la tâche *Formuler_requête* dans la première interface (figure 3) est abstrait par le système de transitions de la figure 4.

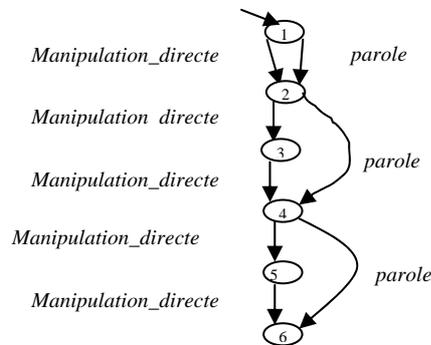


Figure 4 : Système de Transitions Etiquetées de la tâche Formuler_requête

Le système de transitions ainsi obtenu est codé dans le langage d'entrée du contrôleur sur modèle SMV. Certaines des propriétés CARE ont été modélisées par la logique temporelle CTL et vérifiées automatiquement par le model-checker SMV. Seules les propriétés sont présentées, nous ne donnons pas les codes SMV associés aux modèles des systèmes de transitions. Ils ne font que transcrire les représentations des systèmes de transitions.

Propriétés 1. Dans l'interface de type exclusif : les deux modalités *parole* et *manipulation_directe* sont équivalentes pour l'expression de la tâche *Formuler_requête*.

Cette propriété est vérifiée à l'aide de la formule de logique temporelle CTL suivante :

$$\begin{aligned}
 & EG (((etat=1) \textbf{and} (mod=parole)) \\
 & \quad \Rightarrow E((mod=parole) \cup (etat=6))) \\
 \Leftrightarrow & EG (((etat=1) \textbf{and} (mod=manipulation_directe)) \\
 & \quad \Rightarrow E((mod=manipulation_directe) \cup (etat=6)))
 \end{aligned}$$

Les opérateurs *EG*, *U* et *E* signifient de manière informelle, respectivement, « il existe toujours un chemin où », « jusqu'à » et « il existe un chemin où ». La variable *etat* est une variable qui représente le numéro de l'état global du système, qui à son tour regroupe un ensemble de variables d'états du système (non représentées ici). *Mod* est une variable introduite pour représenter le type de modalité utilisée pour réaliser les actions et transiter d'un état à un autre dans le système de transitions.

La formule précédente exprime formellement que l'existence d'une exécution qui démarre de l'état 1 et qui se termine à l'état 6 et dont les transitions sont toutes effectuées par la modalité *parole* est équivalente à l'existence d'une exécution qui démarre de l'état 1 et qui se termine à l'état 6 et dont les transitions sont toutes effectuées par la modalité *manipulation_directe* notée ici *m_directe*.

Propriété 2. Dans l'interface de type alterné : les deux modalités *parole* et *manipulation_directe* sont *complémentaires*. Ceci est vérifié à l'aide de la formule de logique temporelle CTL suivante :

$$\begin{aligned}
 & EG (\\
 & \quad ((etat=1) \textbf{and} ((mod=parole) \textbf{or} (mod=m_directe))) \\
 & \Rightarrow (((mod=parole) \textbf{or} (mod=m_directe)) U (etat=13)) \\
 & \quad) \\
 & \textbf{and} \\
 & \textbf{not} EG (\\
 & \quad ((etat=1) \textbf{and} (mod=parole)) \\
 & \Rightarrow ((mod=parole) U (etat=13)) \\
 & \quad) \\
 & \textbf{and} \\
 & \textbf{not} EG (\\
 & \quad ((etat=1) \textbf{and} (mod=m_directe)) \\
 & \Rightarrow ((mod=m_directe) U (etat=13)) \\
 & \quad)
 \end{aligned}$$

Elle exprime formellement

- l'existence d'une exécution qui démarre de l'état 1 et se termine à l'état 13 et dont les transitions sont effectuées soit par la modalité *parole* soit par la modalité *manipulation_directe* ;
- l'absence d'une exécution dont les transitions sont toutes effectuées par la modalité *parole* ;

- l'absence d'une exécution dont les transitions sont toutes effectuées par la modalité *manipulation_directe*.

CONCLUSION

Dans cet article, nous avons présenté une contribution qui consiste en la définition d'un modèle formel pour la description des interactions multimodales en entrée. La sémantique de ce modèle a été donnée à l'aide des systèmes de transitions étiquetées. Le modèle que nous avons proposé est paramétré et permet la conception de différents types d'IHM3. L'implémentation d'une étude de cas a été faite en utilisant les systèmes de transitions étiquetées et ceci avec une conception ascendante. Nous avons également utilisé la technique du model-checking pour la vérification de propriétés caractérisant les IHM3 issues des propriétés CARE.

Dans un travail futur, nous nous intéressons à l'utilisation de la technique de preuve pour la vérification de propriétés CARE en utilisant une approche fondée sur la preuve comme B ainsi qu'à la conception descendante de l'interface en utilisant le raffinement et la décomposition. Enfin, à l'image de la démarche générique proposée pour la modélisation avec les systèmes de transitions, nous nous efforcerons de généraliser la description des propriétés à partir de la classification CARE.

BIBLIOGRAPHIE

1. Yamine Aït-Ameur, Patrick Girard, et Francis Jambon. « *A uniform approach for the specification and design of interactive systems : the B method* ». In Panos Markopoulos and Peter Johnson, editors, Eurographics Work-shop on Design, Specification, and Verification of Interactive Systems (DSV-IS'98), volume Proceedings, pages 333-352, Abingdon, UK, 1998.
2. Bellik Yacine « Interfaces multimodales : concepts, modèles et architectures » thèse de doctorat d'université 1995.
3. Bier, E., Stone, M. Pier, K. Buxton, W. & DeRose, T. "Toolglass and Magic Lenses: the See-Through Interface", Proceedings of ACM SIGGRAPH'93, ACM Press, pp.73-80, 1993.
4. P. Brun. « XTL : a temporal logic for the formal development of interactive systems », pages 121-139. Springer-Verlag, 1997.
5. Clarke, E. M., Emerson, E. A & Sistla, A. P., « Automatic verification of finite-state concurrent systems using temporal logic specifications », ACM Transactions on Programming Languages and Systems 8(2), 244-263, 1986.

6. Clarke, E.M. & Emerson, E. A. "Synthesis of synchronization skeletons for branching time temporal logic". In logics of programs : workshop, volume 131, York-town Heights, New York, 1981. Springer-Verlag.
7. P. Cohen, M. Johnson, D. McGee, S. Oviatt, J. Pittman, I. Smith, L. Chen & J. Clow , " Quickset: Multimodal Interaction for Distributed Application ". Proceedings of the fifth ACM Int. conference on Multimedia, ACM Press, 1997.
8. Coutaz, J., Nigay, L., Salber, D., Blandford, A., May, J. and Young, R.M. « Four easy pieces for assessing the usability of multimodal interaction: the CARE properties ». In K. Nordby, P.H. Helmersen, D.J. Gilmore and S.A. Arnesen (eds) *Human Computer Interaction: Interact '95*. Chapman and Hall: London. pp. 115-120, 1995.
9. David J. Duke and Michael D. Harrison. " Abstract interaction objects ". Computer Graphics Forum, 12(3) : 25-36, 1993.
10. Duke, D. et Harisson, M. D. « mapping User requirements to implementations ” . In Software Engineering Journal (Vol 10(1), p54-75) 1997.
11. Gregory D. Abowd, Hung-Ming Wang, and Andrew F. Monk. "A formal technique for automated dialogue development". In Gary M. Olson and Sue Schuon, editors, DIS'95, Design of interactive Systems, pages 219-226, Ann Arbor, Michigan, 1995. ACM Press.
12. MacColl et Carrington, D. "testing MATIS : a case study on specification-based testing of interactive systems", FAHCI (1998) , pp.57-69, ISBN 0-86339-7948
13. Z.Manna & A.Pnueli. « the temporal logic of reactive and concurrency systems : specification ». Springer-verlag, 1992.
14. McMillan, K. L., "The SMV system", Technical report, Carnegie Mellon University, 1992.
15. L. Nigay. « Conception et modélisation logicielles des systèmes interactifs : application aux interfaces multimodales », PhD dissertation, Grenoble University, 1994, 315 pages
16. L. Nigay & J. Coutaz. « A Generic Platform for Addressing the Multimodal Challenge », Proceedings of ACM CHI 1995, pp. 98-105.
17. L. Nigay, J. Coutaz, « Espaces conceptuels pour l'interaction multimédia et multimodale », TSI, spéciale Multimédia et collectif, AFCET & HERMES Publ, Vol 15(9), 1996, p. 1195-1225
18. P. Palanque. « Modélisation par Objets Coopératifs Interactifs d'interfaces homme-machine dirigées par l'utilisateur ». Doctorat d'université (phd thesis), Université de Toulouse I, 1992.
19. P. Palanque, A. Schyn. « A model-based for Engineering multimodal interactive systems ». Proceeding of the ninth IFIP TC13 international Conference on Human-Computer Interaction (Interact'2003) sep 1-5, 2003. Zürich, Switzerland.
20. Paterno F. & Faconti G. « On the use of Lotos to describe graphical interaction ». In Monk A., Diaper D. & Harrison MD. People and Computers VII, Proc. HCI'92 Conference, Cambridge University Press, pp. 155-173, 1992.
21. Fabio Paterno, Menica Mazzanotte. « Analysing Matis by Interactors and ACTL ». Amodeus Project Document : system Modelling/WP36, 1994
22. J. Queille & J. Sifakis. « Specification and verification of concurrent systems in Caesar ». In 5th international symp. On programming, Lecture Notes in Computer Science, pages 337-351. Springer Verlag, 1981.
23. P. Roche. « Modélisation et validation d'interface homme-machine ». Doctorat d'université (phd thesis), Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, 1998.
24. A. Waserman. "User software engineering and the design of interactive systems". In 5th IEEE International Conference on Software Engineering, Pages 387-393. IEEE society press, 1981
25. L. S. Marshall. " A Formal Description Method for User Interface ", Phd thesis, University of Manchester, 1986.
26. G. D. Plotkin. "A structural approach to operational semantics". Rapport de recherche DAIMI FN-19 Université d'Arhus Computer Science departement Arhus Danemarke 1981.